What Is Claimed Is:

1. A system for providing a client with access to remote graphics rendering resources, comprising:

a remote rendering control system that receives graphics instructions, generates modified graphics instructions on the basis of said graphics instructions, and outputs said modified graphics instructions to said graphics rendering resources.

2. The system of claim 1, wherein said remote rendering control system comprises a transparent interface to said graphics application, and wherein said transparent interface supports initialization of a graphics rendering session and accommodates client parameters during said graphics rendering session.

3. The system of claim 1, wherein said remote rendering control system comprises a data compression module that compresses said image data prior to sending said image data to said client.

4. The system of claim 1, wherein said remote rendering control system receives image data generated by said graphics rendering resources on the basis of said modified graphics instructions, and sends said image data to said client.

5. The system of claim 1, wherein said remote rendering control system receives graphics instructions from a graphics application program.

6. A method of remote graphics rendering on behalf of a client, comprising the steps of:

(A) initializing a graphics rendering session;

(B) starting a graphics application on the basis of a command from the client;

(C) generating graphics instructions;

(D) imposing client parameters to produce modified graphics instructions;

(E) sending the modified graphics instructions to graphics rendering resources;

(F) rendering graphics on the basis of the modified graphics instructions to produce image data in one or more frame buffers;

(G) reading image data from the one or more frame buffers;

(H) enqueuing the image data; and

(I) transmitting the image data to the client.

7. The method of claim 6, further comprising the step of:

(J) compressing the image data,

performed after step (H) and before step (I).

8. The method of claim 7, wherein steps (F), (J), and (I) are performed in pipeline fashion.

9. The method of claim 8, wherein steps (F), (J), and (I) are asynchronous.

10. The method of claim 6, wherein step (A) comprises the steps of:

(i) performing a client / server handshake;

(ii) receiving a client visual from the client;

(iii) after a user at the client opens a console window at the client and starts the graphics application, opening client and server displays;

(iv) merging the client visual with a server visual to form a merged visual list;

(v) associating the client display with the graphics application;

(vi) overlaying the server visual list with a transparent interface routine;

(vii) enabling the return of a client window to the graphics application;

(viii) enabling the return of an internal context to the graphics application; and

(ix) binding a server context to the server window.

11. The method of claim 10, wherein step (vii) comprises the steps of:

(a) converting the merged visual list into a visual appropriate for the client;

(b) defining the client window;

(c) creating an internal data structure for tracking the displayed location of the client window; and

(d) returning the client window to the graphics application.

12. The method of claim 10, wherein step (viii) comprises the steps of:

(e) converting the merged visual list into a visual appropriate for the server;

(f) creating a server context; and

(g) returning an internal context to the application.

13. The method of claim 10, wherein step (ix) comprises the steps of:

(h) extracting a server context from the internal context;

(i) requesting a window allocation from a session manager; and

(j) associating the server context with a server window.

14. The method of claim 6, wherein step (D) comprises the steps of:

(x) intercepting every function call that includes a visual capability;

(xi) converting the visual capability to a corresponding client visual capability;

(xii) intercepting every reference to a graphics context; and

(xiii) converting every reference to a graphics context to a reference to a graphics context of the client.

15. A computer program product comprising a computer usable medium having computer readable program code that enables remote graphics rendering on behalf of a client, said computer readable program code comprising:

first computer readable program code logic for causing a server to initialize a graphics rendering session;

second computer readable program code logic for causing the server to start a graphics application on the basis of a command from the client;

third computer readable program code logic for causing the server to generate graphics instructions;

fourth computer readable program code logic for causing the server to impose client parameters to produce modified graphics instructions;

fifth computer readable program code logic for causing the server to send the modified graphics instructions to graphics rendering resources;

sixth computer readable program code logic for causing the graphics rendering resources to render graphics on the basis of the modified graphics instructions to produce image data in one or more frame buffers;

seventh computer readable program code logic for causing the server to read image data from the one or more frame buffers;

eighth computer readable program code logic for causing the server to enqueue the image data; and

ninth computer readable program code logic for causing the server to transmit the image data to the client.

16. The computer program product of claim 15, said computer readable program code further comprising:

tenth computer readable program code logic for causing the server to compress the image data.

17. The computer program product of claim 15, wherein said first computer readable program code logic comprises:

(i) computer readable program code logic for causing the server to participate in a client / server handshake;

(ii) computer readable program code logic for causing the server to receive a client visual from the client;

(iii) computer readable program code logic for causing the server to open client and server displays after a user at the client opens a console window at the client and starts the graphics application;

(iv) computer readable program code logic for causing the server to merge the client visual with a server visual to form a merged visual list;

(v) computer readable program code logic for causing the server to associate the client display with the graphics application;

(vi) computer readable program code logic for causing the server to overlay the server visual list with a transparent interface routine;

(vii) computer readable program code logic for causing the server to enable the return of a client window to the graphics application;

(viii) computer readable program code logic for causing the server to enable the return of an internal context to the graphics application; and

(ix) computer readable program code logic for causing the server to bind a server context to the server window.

18. The computer program product of claim 17, wherein said computer readable program code logic (vii) comprises:

(a) computer readable program code logic for causing the server to convert the merged visual list into a visual appropriate for the client;

(b) computer readable program code logic for causing the server to define the client window;

(c) computer readable program code logic for causing the server to create an internal data structure for tracking the displayed location of the client window; and

(d) computer readable program code logic for causing the server to return the client window to the graphics application.

19. The computer program product of claim 17, wherein said computer readable program code logic (viii) comprises:

(a) computer readable program code logic for causing the server to convert the merged visual list into a visual appropriate for the server;

(b) computer readable program code logic for causing the server to create a server context; and

(c) computer readable program code logic for causing the server to return an internal context to the application.

20. The computer program product of claim 17, wherein said computer readable program code logic (ix) comprises:

(a) computer readable program code logic for causing the server to extract a server context from the internal context;

(b) computer readable program code logic for causing the server to request a window allocation from a session manager; and

(c) computer readable program code logic for causing the server to associate the server context with a server window.

21. The computer program product of claim 15, wherein said fourth computer readable program code logic comprises:

(i) computer readable program code logic for causing the server to intercept every function call that includes a visual capability;

(ii) computer readable program code logic for causing the server to convert the visual capability to a corresponding client visual capability;

(iii) computer readable program code logic for causing the server to intercept every reference to a graphics context; and

(iv) computer readable program code logic for causing the server to convert every reference to a graphics context to a reference to a graphics context of the client.